Character Recognition with Fuzzy Features and Fuzzy Regions

Erick Robertino Mertooetomo and Jianhua Chen
Department of Computer Science, Louisiana State University, Baton Rouge, Louisiana.

Abstract
We propose a method for character recognition using fuzzy features and fuzzy regions in a neural network. The method is robust to noise and distorting, scaling, and shifting of the patterns within their pixel frames, yet it is mathematically simple. The fuzzy neural network presented in this paper consists of three layers: A layer for feature extraction, for regional emphasis of features, and for classification. We extract features from regions of the characters in which they are most likely to occur. To make the system robust, these regions are fuzzified, giving higher weight to areas where the features are most likely to occur and lower to areas where the features are rare. Sample patterns from the literature have been used for training of the network to obtain the minimal set of distinguishing features with their associated measures and to determine the optimal slopes of these linear regions. The network has been tested using patterns from the literature. Its performance is comparable for distorted and noisy patterns and superior for shifted, partial, and down-scaled samples.

1. Introduction

Automatic character recognition has been an active area of research in recent years, with numerous methodologies and experimental systems reported in the literature. In spite of the success in this area, it seems that most of the existing approaches to character recognition are either sensitive to noise [1], scaling and shifting [2] (shown in this paper), and distortions [3] of the patterns, or they require complexity in order to be robust with respect to the problems mentioned above [4, 5]. Towards the solution of this robustness-complexity dilemma, we propose a fuzzy neural network approach, which is quite robust yet simple. Our work is related to the method in [2] which uses a very simple fuzzy neural network based on fuzzified template matching for character recognition. The main drawback of the approach in [2] is the system's sensitivity to scaling and shifting. We address this problem by using fuzzy features and fuzzy regions. The proposed fuzzy neural network (called feature net) classifies patterns based on features extracted from regions in which they are most likely to occur. The features are fuzzified using triangular membership functions and the regions using linear functions. This generalizes the work in [6] in which only fuzzy borders are considered. We will first describe the feature net's architecture, then the selection of features, and the methods to assign various weights of the neural network. Finally, we will present the experimental results and compare them with [2].

2. Feature net architecture

Patterns consist of 16x16 pixels with gray values (normalized) in the range 0 to 1. The lowest row number (iL) and column number (jL) and the highest row number (iH) and column number (jH) at which a nonzero gray level pixel occurs, enclose the character frame. The character's base is defined as (b = jL−jH), height (h = iH−iL), and the diagonal length (D = iH−iL + jH−jL). Feature net consists of three layers of neurons as shown in Fig. 1. There are \( n_R \times n_C \) extract neurons, where \( n_R \) is the number of rows and \( n_C \) the number of columns in the pixel frame. If \( n_F \) is the number of features, then there will be \( n_F \) neurons in the feature layer. There will be as many class neurons as there are classes of patterns.

The extract neurons extract features from the matrix of normalized gray levels originating from the pixel frame, by applying feature weights to these gray levels. Each extract neuron extracts the features present at its corresponding position in the pixel frame. If the feature is the horizontal stroke, then the net input to the extract neuron at position \((p,q)\) is the score for the horizontal strokes emanating from that position in the pixel frame. There are as many states in the extract neurons as there are features: One state for each feature. Let \( f_{ij}^{b,p,q} \) be the feature weight for the \( ij \) th pixel for the \( k \) th feature and the \( p,q \) th extract neuron, \( y_{ij}^{k,p,q} \) the normalized gray level of the \( ij \) th pixel, \( u_{k}^{b,p,q} \) the net input to the \( p,q \) th extract neuron for the \( k \) th feature, \( s_{k}^{b,p,q} \) the \( k \) th state (output) associated with the \( k \) th feature of the \( p,q \) th neuron, \( n_{n} \) the number of pixels in the pixel frame with a nonzero gray level (a level above a chosen threshold). \( k \) ranges from 1 to \( n_{n} \).
Maximum class membership

Fig. 2. Feature-measure degree of membership.

Since some feature-measure combinations have more distinguishing power than others, \( \mu_{k,m} \) is given a weight, \( w_{k,m} \). \( w_{k,m} \) is independent of class. For each class, all the sample’s feature-measure degree of memberships multiplied by their weights are added to give a class degree of membership (\( \mu_C \)):

\[
\mu_C = \frac{\sum_k \sum_m w_{k,m} \mu_{k,m} (k=0 \text{ to } n_F-1, m=0 \text{ to } M_k-1)}{\sum_k M_k (k=0 \text{ to } n_F-1)}
\]

The class with the highest class degree of membership is chosen for the sample classified. Aggregation of the feature scores depends on the type of measure used, e.g. the average, maximum, or minimum of the pixel feature scores. Let \( u_{k,m} \) be the net input to and \( s_{k,m} \) the state (output) of the \( k \)th feature neuron for the \( m \)th measure (\( m \)th state), \( y_{i,j,k} \) is the \( k \)th feature score at the \( i,j \)th pixel (the output of the \( i,j \)th extract neuron for the \( k \)th feature), \( r_{i,j,k} \) is the region weight assigned to the \( i,j \)th pixel for the \( k \)th feature, \( k \) ranges from 0 to \( n_F-1 \), \( m \) from 0 to \( M_k-1 \). We then have:

\[
u_{k,m} = g^{k,m}[y_{i,j,k} \mid i=0 \text{ to } n_R-1, j=0 \text{ to } n_c-1]
\]

giving the following forms for particular measures:

- \( u_{k,m} = \frac{\sum_j y_{i,j,k}}{n_N} \) for the average feature score over the entire pixel frame with no regional emphasis, denoted by Ave.

- \( u_{k,m} = \frac{\sum_j r_{i,j,k} y_{i,j,k}}{n_N} \) for the average feature score over the entire pixel frame with regional emphasis, identified by bold region names not preceded by Max or Min (e.g. Equator Meridian intersection).

- \( u_{k,m} = \max(r_{i,j,k} y_{i,j,k} \mid i=0 \text{ to } n_R-1, j=0 \text{ to } n_c-1) \) for the maximum feature score over the pixel frame with regional emphasis, designated by bold region names preceded by Max (e.g. Max Northwest).

3. Selection of features

By inspecting the exemplars, we obtained the set of features and their associated measures used for distin-
### TABLE I
Minimal set of distinguishing feature-measure combinations

<table>
<thead>
<tr>
<th>Feature</th>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>EW, EW1</td>
<td>Ave Northern</td>
</tr>
<tr>
<td></td>
<td>Southern</td>
</tr>
<tr>
<td></td>
<td>Equator</td>
</tr>
<tr>
<td></td>
<td>Equator Meridian intersection</td>
</tr>
<tr>
<td></td>
<td>North Gap</td>
</tr>
<tr>
<td></td>
<td>South Gap</td>
</tr>
<tr>
<td></td>
<td>East Deep Gap</td>
</tr>
<tr>
<td>NS, NS1</td>
<td>Ave Eastern</td>
</tr>
<tr>
<td></td>
<td>Western</td>
</tr>
<tr>
<td></td>
<td>Meridian</td>
</tr>
<tr>
<td></td>
<td>Equator Meridian intersection</td>
</tr>
<tr>
<td></td>
<td>East Gap</td>
</tr>
<tr>
<td></td>
<td>West Gap</td>
</tr>
<tr>
<td></td>
<td>North Deep Gap</td>
</tr>
<tr>
<td></td>
<td>South Deep Gap</td>
</tr>
<tr>
<td>NE, NE1</td>
<td>Ave Northwest</td>
</tr>
<tr>
<td></td>
<td>Southeast</td>
</tr>
<tr>
<td></td>
<td>Northeast Diagonal</td>
</tr>
<tr>
<td></td>
<td>Center</td>
</tr>
<tr>
<td>NW, NW1</td>
<td>Ave Northwest Diagonal</td>
</tr>
<tr>
<td></td>
<td>Center</td>
</tr>
<tr>
<td>Northwest Corner</td>
<td>Max Northwest</td>
</tr>
<tr>
<td>Southwest Corner</td>
<td>Max Equator Western intersection</td>
</tr>
</tbody>
</table>

Fig. 3. Kwan and Cai exemplars.


Fig. 4. Kwan and Cai distorted, noisy, and scaled samples.

Distinguishing one class from another. We chose the features which occur in the highest number of exemplars. The most appropriate measures for these features were obtained by considering in which regions these features
most likely occur. The following features have been implemented. Note that the linear features are in fuzzy-crisp pairs. The distinction between a fuzzy feature and its crisp counterpart lies in the way the feature weight is defined, as explained in section 4.

- The fuzzy (denoted by EW) and crisp (EW1) horizontal stroke:
- The fuzzy (NS) and crisp (NS1) vertical stroke:
- The fuzzy (NE) and crisp (NE1) northeast diagonal stroke:
- The fuzzy (NW) and crisp (NW1) northwest diagonal stroke:
- The northwest corner (point at which horizontal and vertical legs meet as in the next figure):
- The southwest corner (intersection point as in the following figure):

The minimal set of feature-measure combinations, shown in TABLE 1, was obtained using as training samples the 36 Kwan and Cai exemplars [2], shown in Fig. 3, and the 40 Kwan and Cai samples with noise, scaling, and distortion, shown in Fig. 4. The regions (from measures) appearing in the minimal set are described in [7].

4. Feature weights

The use of crisp linear feature weights is equivalent to taking the histogram of the character in the direction specified e.g. horizontal direction. If the i,j th pixel lies along the stroke as viewed from the p,q th pixel, it gets a weight of 1, otherwise a weight of 0. This is illustrated in Fig. 5 for the horizontal stroke and a specific (p,q). The

\begin{align*}
    f(i,j) &= 1 \text{ if } i = p \\
    &= 0 \text{ otherwise}
\end{align*}

To incorporate information about how far the i,j th pixel lies from the p,q th pixel, a lower weight is given to the i,j th pixel the farther it is from the p,q th pixel in the fuzzy linear features. The distance between (i,j) and (p,q) is divided by \( S_k \), the longest stroke possible in the stroke direction given the size of the pixel frame. The distance (street distance) is defined as:

\[ d((p,q),(i,j)) = |p-i| + |q-j| \]

The fuzzy weights are assigned as follows:

\[ f(i,j) = 1 - d((i,j),(p,q))/S_k \text{ if } (i,j) \text{ is on stroke from } (p,q) \]

\[ = 0 \text{ otherwise} \]

where \( S_k = d((0,0),(nR-1,nC-1)) = nC-1 \) for the horizontal stroke, \( S_k = d((0,0),(nR-1,0)) = nR-1 \) for the vertical stroke, and \( S_k = d((nR-1,0),(0,nC-1)) = d((0,0),(nR-1, nC-1)) = nC+nR-2 \) for the northeast and northwest diagonal strokes. The feature weights for the horizontal stroke from a specific (p,q) are shown in Fig. 6.

![Fig. 6. Horizontal stroke fuzzy feature weights.](image)

5. Region weights

Regional emphasis is obtained by giving pixels within a region a higher weight than pixels outside the region. Let the weight, associated with region emphasizing, assigned to pixel i,j be denoted by \( r_{ij} \). In the literature until now, regions have been crisp. Crisp regions can be inadequate in cases where there are distortions and shifting. Therefore, we generalized the work in [6] by considering the more general fuzzy regions in this work.

In our method, after determining the size of the crisp emphasis region (e.g. by inspecting the exemplars and the sample patterns), we proceed by using linear or bilinear functions to compute the fuzzy region weights (\( r_{ij} \)). A simple way of assigning \( r_{ij} \) is to use the (smallest) street distance of pixel (i,j) to the pixel or collection of pixels with the lowest region weight. The user should choose the pixel(s) to be given the lowest region weight, which would designate the pixel(s) with the largest street distance away as the pixel(s) with the highest region weight. Let \( L \) denote the pixel(s) with the lowest region weight.
weight and \( H \) the pixel(s) with the highest region weight. So, \( r_{ij} = \text{function}(d[(i,j),L]) \). To keep the mathematics simple, linear functions have been chosen for function: \( r_{ij} = ad[(i,j),L] \), where \( a \) is the slope of the linear region. An example of single linear regions is shown in Fig. 7.

\[
\begin{align*}
\text{Fig. 7. Single linear fuzzy northern region.}
\end{align*}
\]

Single linear regions are not sufficient in cases where the region of emphasis needs stronger emphasis with respect to the non-emphasis region. Increasing the slope does not change the degree of emphasis. In such cases, we used a bilinear region, Fig. 8, which has two slopes: The

\[
\begin{align*}
\text{Fig. 8. Bilinear fuzzy northern region.}
\end{align*}
\]

non-emphasis region’s slope is still left at 1, while a higher slope \( (a) \) is given to the emphasis region. Let \( B \) denote the border pixels separating the emphasis region from the non-emphasis region and \( b \) the fraction of \( d[L,H] \) belonging to the non-emphasis region: \( b = d[L,B]/d[L,H] \). Note that by convention, the distance between two sets of pixels \( d[s_1,s_2] \) is the shortest distance between any two pixels, one belonging to \( s_1 \) and one to \( s_2 \). \( d[L,H] \) is a characteristic length. The region weights are assigned as follows:

\[
\begin{align*}
\text{If } d[L,(i,j)] < bd[L,H] \\
\quad r_{ij} = d[L,(i,j)] \\
\text{else} \\
\quad r_{ij} = bd[L,H] + ad[(i,j),B] \\
\quad = bd[L,H] + a[d[(i,j),L]-d[L,B]] \\
\quad = bd[L,H] + a(d[(i,j),L]-bd[L,H])
\end{align*}
\]

Note that the resulting formulas for \( r_{ij} \) do not require the coordinates of the border pixels (B). The characteristic length \( (d[L,H]) \), the fraction of the characteristic length belonging to the non-emphasis region \( (b) \), and the coordinates of the point(s) with the lowest weight \( (L) \) are all that needs to be known besides \( (i,j) \).

We used a set of training samples to determine the optimum values of \( a \) and \( b \), that give the highest recognition rate for the set. It is an iterative approach with \( a \) and \( b \) being the independent variables and the recognition rate the dependent variable. First, \( b \) was fixed by inspection of the exemplars and determining the region in which the feature most likely occurs [7]. An initial guess of 1 (single linear region case) and another arbitrary value was chosen for \( a \). The recognition rate for these cases was then determined. The optimum value of \( a \) was then found by interpolation and/or extrapolation.

Intersection regions are obtained by multiplicative superimposition. The region weights for the intersection region of regions A and B are given by: \( r_{ij}^{AB} = r_{ij}^A \cdot r_{ij}^B \)

6. Experiments

We have implemented our method using C++ on the DEC 5000 platform. The details of the implementation (parameter tuning, etc.) can be found in [7]. A series of experiments has been performed to validate the methodology proposed here. In each experiment, the performance of the feature net was comparable to or exceeded Kwan and Cai’s net. Five experiments have been conducted:

- The 36 Kwan and Cai exemplars, shown in Fig. 3. Both feature net and Kwan and Cai’s net recognized all exemplars.
- The shifted patterns, obtained by shifting the 36 exemplars 1, 2, and 3 pixels in the Up (U), Down (D), Left (L), Right (R), Up-Left (UL), Down-Left (DL), Up-Right (UR), Down-Right (DR) directions. An example of the 1 and 2 pixel shifts is given in Fig. 6 of [2]. Both feature net and Kwan and Cai’s net recognized all 1 pixel shifts of the exemplars. As expected from the design, feature net recognized all the shifted samples (1,2,3 pixel shifts). Kwan and Cai’s recognition rate deteriorates as the shifts become larger. The recognition rate is 100% for 1 pixel shifts, 88.2% for 2 pixel shifts, and 57.3% for 3 pixel shifts, as can be seen from TABLE IV. The lowest recognition rate for 2 pixel shifts in a particular shift direction (DR) is 75.0%, shown in TABLE II, and for 3
Feature net is not affected by pixel shifts at all, while Kwan and Cai's net is limited by its local pixel to pixel comparison approach.

- The Kwan and Cai's distorted, noisy, and scaled samples shown in Fig. 4. Feature net’s recognition rate is comparable to Kwan and Cai’s. Feature net recognized all the 40 Kwan and Cai’s distorted, noisy, and scaled samples except the SH C sample (recognized as G) and the AP C sample (recognized as B), yielding a recognition rate of 95.0%. Kwan and Cai’s net did not recognize only one. The HS H sample was classified ambiguously as H and N, giving a recognition rate of 97.5%. So, the recognition rates are comparable.

- The partial samples shown in Fig. 9. Feature net recognized all four of the partial samples (100%), while Kwan and Cai’s net recognized none (0%). Note that the partial samples with some pixels deleted still maintain the overall shape of the original exemplar, and thus should be recognized.

- The down-scaled samples shown in Fig. 10. The exemplars C, H, Z, and 4 have been scaled down even further than the SM samples of Kwan and Cai [2] to test the lower-scale limit. Feature net recognized all down-scaled samples (100%), while Kwan and Cai’s net recognized only one (25%), the down-scaled 4 sample.

Overall, feature net performs better than Kwan and Cai’s net as can be seen from the recognition rates shown in Table IV.

| TABLE IV | Overall recognition rates(%) |
|---|---|---|---|---|---|---|---|---|---|
| | One pixel shifts | Two ... | Three ... | Distorted, noisy, and scaled samples | Partial | Down-scaled |
| Feature net | 100 | 100 | 0 | 100 |
| Kwan and Cai’s net | 83.3 | 86.1 | 91.7 | 86.1 |

7. Conclusion

It is possible to have a mathematically simple pattern recognition system that is robust to noise and distorting, scaling, and shifting of the patterns within their pixel frames. We employ fuzzy features to measure the degree to which features are present at each pixel of the pixel frame and fuzzy regions to give emphasis to regions of the patterns where the features are most likely to occur. The performance of feature net is comparable to Kwan and Cai’s net for the 40 Kwan and Cai’s distorted, noisy, and scaled samples, but clearly superior for the shifted, partial, and down-scaled samples. Future work should focus on making feature and region determination automatic.

8. References